LLNL-TR-670049

LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

NA-22 2014 Visible Data

S. Manay

April 27, 2015

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

NA-22 2014 Visible Data

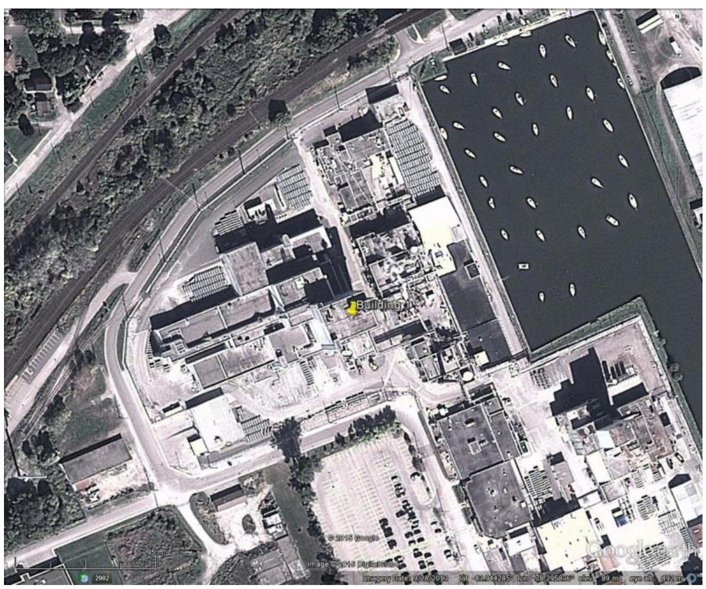
Dataset 1: Visible Imagery of Building 1, Canada, Sept 2014

Siddharth Manay
Lawrence Livermore National Laboratory
28 Apr 2015 (v1)
manay2@llnl.gov

LLNL-TR-670049
This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

Access or use of data on this disk is subject to the terms and conditions outlined in the Data Sharing Agreement between City College of New York (Prof. George Wolberg) and U.S. DoE, NA-22 (Victoria Franques, Proliferation Detection Program) dated 20 Apr 2015.

This data was obtained during a DoE data collect using the PAHSI instrument in Canada in Sept 2014. The instrument collected black-and-white visible imagery from a framing camera on 2014-09-19. This imagery is a subset of the entire data collect, focusing on a single building.



This imagery is intended to allow an initial evaluation of the feasibility of computer vision methods. Future data transfers can be done using different image formats, bitrates, or compression methods. More data is available if an initial evaluation shows this dataset is incomplete or more test data is desired; for instance, imagery is available for all buildings on the site, and similar imagery was captured on other days.

Data Files

Each image is stored in a single file. Each image has several files of associated metadata, as shown in the Table 1. Each file associated with an image has the same base filename, the extension will indicate what the file contains.

Type	Extension	Location	Contents
Image	.tif	Vis/	Image in grayscale 16bit. Stored with Lossless Deflate compression.
GPSINS	.gpsins	GPSINS/	GPSINS (lat, lon, alt, roll, pitch, yaw) and roll compensation mirror information. Binary.
Gcp	.gcp	Vis/Geo/	Approximate geographic position (lat, lon) for each corner of the image. ASCII
Timestamp	.time	Vis/	Time of image acquisition as POSIX time and a human-readable string. ASCII

Table 1: Metadata files associated with each image

The filenames embed some metadata about the file. Table 2 shows an example filename with metadata labeled.

S1__PH__2.context.2014-09-19T15-58-58.005830	
Red	Flightline (aircraft pass) identifier. Several passes may share the same identifier if they were repeats of the same geographical collection.
Green	Camera label. In this case, all images are from the “context” imager. Note for other cameras the label is between the Timestamp and Serial image number.
Blue	Timestamp, GMT. YYYY-MM-DDThh-mm-ss
Purple	Serial image number.

Table 2: Filename format

Camera Information

The visible imager (camera) mounted on Pahsi II is an Illunis RMV-16070 Grayscale.
http://www.illunis.com/camera_selector/ccd_cameras/7.4_cameras/7.4_cameras.html

Pixel Pitch	7.4 µm
Image Dimensions	4864 x 3232 pixels
Focal Length	18 cm

Table 3: Camera Specifications.

GPSINS and Roll Compensation Mirror

The PAHSI II sensor system includes a GPSINS system and a Roll Compensation Mirror.

The GPSINS system is commercially available hardware, an Applanix POS AV 410 V5. More information about the GPSINS system can be found at:
http://www.applanix.com/media/downloads/products/specs/posav_specs_1212.pdf

The GPSINS system tracks the position and orientation of the sensor's "baseplate", so that the vector normal to the baseplate (pointing out the bottom of the aircraft) is known with high accuracy. The visible imager is mounted to the baseplate.

The system also has a Roll Compensation Mirror; the visible imager looks through this mirror at the ground. The mirror is attached to a turning stage and a control loop designed to keep the imager line-of-site (LoS) pointing at the desired angle (in earth coordinates, e.g. true nadir or at a 40degree oblique angle). See Figure 1 for a schematic of the camera, INS, and roll compensation system.

The mirror system is custom-designed; please contact me if more information is needed.

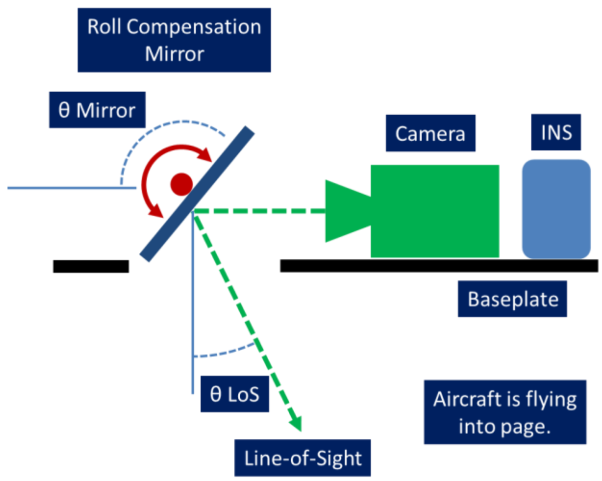


Figure 1: Diagram of visible camera, INS, and Roll Compensation Mirror system.

Appendix: Code Fragments to read GPSINS files

```
% Code fragments describing data in the .gpsins binary file.
% Siddharth Manay
% 23 Apr 2015

%
% Fragment 1: Reading a GPSINS file
%

%  fn - filename (with path)
%
% frameTime - times for frames in HSI file in GPS format (milliseconds since
Sunday Midnight GMT)
% gpsTime - times for GPS records (milliseconds since Sunday Midnight GMT)
% gpsArr - 6 x numGpsRecord array with GPSIMU data.  Rows:
%   1 - lat (deg)
%   2 - lon (deg)
%   3 - alt
%   4 - roll  (deg)
%   5 - pitch (deg)
%   6 - yaw   (deg)
% mirrorTime - times for the mirror records (milliseconds since Sunday
%   Midnight GMT)
% mirrorArr - 1 x numMirrorRecord array with mirror data.  Rows:
%   1 - viewing angle (deg; positive numbers move the LoS towards the RIGHT
%   wing).

% Open file
fid=fopen(fn,'rb'); %assumes fn is a filename string

% Read header information
idStr = fread( fid, [1,32] ,'*char*1' );  % This should contain 'LLNL-PRS-
GPS-v2'
nf=fread(fid,[1,1],'uint64'); % number of frames from the spectrometer
nGps=fread(fid,[1,1],'uint64'); % number of Gps records
nMir=fread(fid,[1,1],'uint64'); % number of mirror records
% NOTE: for visible context imagery, nf, nGps, and nMir will all be 1.
% For HSI (linescanner) imagery, nf, nGps, and nMir will all be different
% numbers (as the three devices collect data at different rates.

% Read frame timestamp from spectrometer
frameTime=fread(fid,[nf,1],'uint64'); % frame time stamps

% Read GPS data:
gpsDat=fread(fid,[7, nGps],'*uint64'); % gps times
gpsTime = double(gpsDat(1,:)); %values were uint64 in file, so convert the
uint64 values to double.

nGps0 = size( gpsDat, 2 );
gpsArr = reshape( typecast( reshape( gpsDat(2:7,:), 1, []), 'double' ), 6,
nGps0 ); %values were double in file, so *cast* the uint64 to double.
% ^ unfortunately, typecast only works on vectors, so reshaping is needed
% columns of gpsArr now contain:
% TODO
% The ith row in gpsArr is the data collected at gpsTime[i]
```

```
% Read the Roll Compensation Mirror Data
mirDat=fread(fid,[2,nMir],'*uint64'); % mirror array.
mirrorTime = double(mirDat(1,:)); %values were uint64 in file, so convert the
uint64 values to double.
mirrorArr = typecast( mirDat(2,:), 'double' ); %values were double in file,
so *cast* the uint64 to double.




%
% Fragment 2: Using the mirror offset:
%
% Assuming...
% - you are working on the i^th row of data
% - the angle offsets between the INS and the LoS (offroll, offpitch, offyaw)
have been computed
% - mirror data in mirrorArr has been interpolated onto the same time
%   domain as gpsArr (or they both have been interpolated onto the same time
%   domain)
roll=gpsArr(i,4) + offroll;
pitch=gpsArr(i,5) + offpitch;
yaw=gpsArr(i,6) + offyaw;
lat=gpsArr(i,1);
lon=gpsArr(i,2);
% Add the view angle offset due to the mirror.
roll = roll - mirArrInterp(i); %Sign has been checked... use a negative so
that positive mirror angle moves FoV to the AC's right.
```